비디오 데이터의 내용 기반 검색과 브라우징을 위한 유동 속성 트리 및 부분 결과 행렬의 이용 방법 연구

성인용[†] · 이원석[††]

요 약

연속된 비디오 스트림에는 다양한 정보가 서로간의 명확한 경계 구분 없이 표현되며 비디오 신의 의미는 여러 추상화 단계로 해석되어질 수 있다. 또한 비디오에 대한 기술은 사용자의 목적에 따라 각각 다르게 표현될 수 있다. 따라서 비디오 데이터에 대한 내용 기반 검색에서는 사용자가 특정 장면에 대한 내용을 유동적으로 기술할 수 있도록 지원하여야 함과 동시에 다양한 사용자가 기술한 내용이 일관되게 유지되어야 한다. 본 논문은 관계형 모델과 객체 지향 모델과 같은 기존의 데이터베이스 모델에서 효과적인 내용 기반 검색 및 브라우징 방법을 제안한다. 유동적으로 정의된 속성과 속성값은 트리 구조의 사전 형태로 구조화되며 비디오 데이터에 대한 기술은 고정 데이터베이스 스키마에 저장된다. 또한, 본 논문에서는 사용자의 효과적인 비디오 브라우징 작업을 지원하기 위한 브라우저를 제안한다. 사전 브라우저(dictionary browser)는 사용자의 질의 표현과 의미 기술 작업을 단순화시키고, 결과 브라우저(result browser)는 사용자가 질의 조건들의 다양한 조합에 대한 질의 결과를 분석할 수 있도록 지원한다.

A Study on Flexible Attribute Tree and Partial Result Matrix for Content-based Retrieval and Browsing of Video Data

In-Yong Sung[†] and Won-Suk Lee[††]

ABSTRACT

While various types of information can be mixed in a continuous video stream without any clear boundary, the meaning of a video scene can be interpreted by multiple levels of abstraction, and its description can be varied among different users. Therefore, for the content-based retrieval in video data, it is important for a user to be able to describe a scene flexibly while the description given by different users should be maintained consistently. This paper proposes an effective way to represent the different types of video information in conventional database models such as the relational and object-oriented models. Flexibly defined attributes and their values are organized as tree-structured dictionaries while the description of video data is stored in a fixed database schema. We also introduce several browsing methods to assist a user. The dictionary browser simplifies the annotation process as well as the querying process of a user while the result browser can help a user analyze the results of a query in terms of various combinations of query conditions.

1. 서 론

컴퓨터와 통신 그리고 데이터 압축 기술의 발달로 비디오 데이터를 이용한 다양한 서비스가 가능하게 되었다[1,2]. 다른 데이터 타입과 비교할 때 비디오 데이터는 압축되어있다 하더라도 크기가 매우 크고 [3] 실시간에 연속적으로 접근해야 하므로 비디오 데이터의 관리가 어렵다. 또한 연속된 비디오 스트림에

[†] 현재 동양시스템즈 KM사업본부
[††] 연세대학교 컴퓨터과학과

는 다양한 정보가 서로간의 명확한 경계 구분없이 표현된다. 그러나 비디오 데이터는 복합적인 정보를 용이하게 표현할 수 있으므로 비디오 데이터의 사용이 증가될 것으로 예상된다. 효과적인 내용 기반 검색을 지원하기 위해서는 비디오 파일에 나타나는 임의의 부분에 대한 정보를 적절하게 기술할 수 있는 *기술 단위(unit of description)*로 정의할 수 있어야 하고 이것은 비디오 파일에 있는 부분들을 물리적으로 재구성하는 것이 아니라 물리적으로는 분리되어 있지만 논리적으로는 하나의 연속된 비트 스트림으로 관리될 수 있어야 한다.

비디오 데이터에 대한 정보는 두 가지 유형이 있는데 하나는 비디오 파일의 부분에 나타나는 내용에 대한 정보이고, 다른 하나는 비디오 파일 자체 혹은 그 비디오 파일에 나타나는 개체에 관련된 일반적인 정보이다. 전자는 비디오 파일에 나타나는 개체의 위치와 움직임의 형태 및 의미를 표현하는 정보이며 후자는 비디오 파일의 물리적인 성질과 개체에 대한 일반적인 특성을 표현하는 정보이다. 이 차이를 구분하기 위해서 전자를 동적 정보(*dynamic information*), 후자를 정적 정보(*static information*)라고 정의한다. 비디오 데이터베이스에서 내용 기반 검색에 대한 대부분의 연구는 동적 정보를 효과적으로 표현하는 방법에 초점을 두었지만 일반적으로 질의는 정적 정보와 동적 정보의 조합으로 표현될 수 있으므로 정적 정보와 동적 정보의 유기적인 관계를 표현할 수 있어야 한다.

비디오 데이터베이스 시스템에서 내용 기반 검색의 효율성은 기술 단위의 구조뿐만 아니라 동적 정보의 표현 방법에 달려있고 구축하고자 하는 응용 분야의 필요에 따라서 표현하는 동적 정보의 형태가 변할 수 있다. 따라서 특정 응용 분야에서 요구되는 동적 정보의 형태를 고정하는 것이 더 효율적일 수 있다. 하지만 이런 방법은 응용 분야에 따라 모델링 요소가 변하므로 일반적인 비디오 데이터베이스 모델로 사용하는 데에는 한계가 있다. 따라서 대부분의 연구는 기존 데이터베이스 관리 시스템(DBMS)에서 사용하는 속성의 개념없이 다양한 형태를 자유롭게 모델링할 수 있는 키워드(keyword) 방법을 사용한다[4-6]. 이 방법은 키워드를 저장할 수 있는 간단한 구조로 구현이 가능하지만 질의의 검색 결과가 주석(annotation) 처리에서 사용된 단어와 질의를 표현할 때 사용되는 단어의 선택에 따라 상당히 민감하게 변하는 단점을 갖는다.

속성의 개념을 지원하는 방법에서는 신의 의미를 기술하는 속성들의 집합을 고정된 데이터베이스 스키마(schema)로 표현하는 방법[7-10]과 스키마 없이(schemaless)[11,12] 동적으로 구성하는 형태[11,12]로 분류할 수 있다. 하지만, 동적 정보를 고정 스키마로 표현하는 데에는 다음과 같은 단점이 있다. 첫번째는 신에 나타나는 개체의 움직임에 대한 수많은 유형이 있을 수 있기 때문에 움직임의 형태를 미리 고정시키는 방법은 일반적인 해결책이라 할 수 없다. 두번째는 비디오 데이터베이스에 저장된 신의 종류와 개체의 양이 방대할 수 있고 사용자 관점의 변화에 따라 개체와 신의 중요도가 변할 수 있기 때문에 모델링되는 개체 혹은 신의 종류를 고정하는 것은 좋은 방법이 아니다. 세번째는 동일한 동적 정보에 대한 기술은 사용자의 관점에 따라 표현이 상이할 수 있다는 점이다. 이러한 단점으로 모델링 요소의 집합을 고정하여 정의하는 것은 비디오 데이터베이스가 특정한 용도로 설계되지 않는한 거의 불가능하다.

비디오 데이터베이스 시스템의 이런 요구에도 불구하고 스키마없이 속성을 관리하는 데이터베이스 관리 시스템은 없으므로 대부분의 연구는 서로 다른 질의 언어와 인덱스 구조를 갖는 고유의 비디오 데이터 모델을 제공하고 있다. 이러한 어려움을 극복하기 위해 본 논문은 기존의 데이터베이스 모델에서 유동적으로 정의되는 속성과 속성값의 집합을 효율적으로 관리하는 방법을 제안한다. 또한, 비디오 데이터베이스에서는 기존 데이터베이스 시스템과는 달리 사용자가 질의 결과를 활용하기 위해서는 질의 결과 비디오 데이터를 재생해야 하는데 질의 결과의 수가 상당히 많을 경우 일일이 모든 비디오를 재생한다는 것은 불가능하다. 일반적으로 내용 기반 검색을 위한 질의는 정확한 매칭(matching)보다는 유사한 조건을 갖는 데이터를 추출하는 특성을 갖는다. 따라서, 질의를 구성하는 각 조건에 대한 예상 결과수를 질의 처리 전에 사용자에게 제시함으로써 사용자의 질의 표현 작업을 효과적으로 지원할 수 있다. 본 논문에서는 이러한 기능을 지원하는 결과 브라우저를 제안하여 사용자가 질의 조건들을 재구성할 수 있도록 지원한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존

연구에 대해 분석하고, 3장에서는 선행연구[21]을 기반으로 한 비디오 데이터베이스를 구성할 수 있는 다양한 유형의 속성들을 분류하고 이들을 기존의 데이터베이스 모델에 적용하는 스키마 구조를 제안한다. 4장에서는 스키마 구조에 대한 다양한 질의 형태를 기술하며, 5장에서는 유동적으로 정의된 속성에 대한 내용 기반 브라우징 기법과 질의 결과 브라우징 방법을 제안하고, 6장에서는 본 논문의 결론을 맺는다.

2. 기존 연구

비디오 데이터베이스에 대한 내용 기반 검색의 지원을 위한 다양한 방법의 연구가 진행되고 있다. [4]에서는 비디오 파일과 심볼 객체(symbol object)를 각각 비디오 클래스 계층(class hierarchy)과 심볼 클래스 계층(symbol class hierarchy)으로 구성한다. 이런 계층 구조에 기초하여 기술된 비디오 정보는 *video_sysmbol_object* 테이블에서 유지되고 질의는 CVQL이라는 질의 언어로 표현된다. 심볼 객체의 공간적 또는 시간적 관계는 미리 정의된 시공간 함수로 정의된다. [7]에서는 *topical categories*라고 하는 클래스 계층을 사용하여 도메인의 특정 정보를 표현한다. 키워드는 비디오 정보를 기술하는데 사용되며 도메인의 특정 스키마에 저장되는 반면에 질의는 자유 문구(free-text) 형식으로 표현된다. 위의 두 가지 방식은 주석과 인덱싱을 위해 응용 도메인의 특정 정보를 사용하기 때문에 제한된 응용 분야에 적합하다. 또한 의미 단어들에 대한 의미 사전 트리의 크기가 커질 경우 트리 구조에 대한 브라우징 방법을 고려하지 않았다.

[5]에서는 텔레비전 뉴스의 내용 기반 검색을 위해서 문자 방송을 이용하여 뉴스 정보를 기술한다. 이 방법에서 질의는 키워드로 표현하고 질의 결과는 질의문과 문자 방송의 텍스트간의 매칭 정도를 나타내는 매칭 스코어(matching score)로 정렬하여 순차적으로 표시된다. 키워드와 자유 문구 형식으로 질의를 표현하는 방식은 개체의 위치, 크기, 색깔과 같은 하위 레벨의 정보를 표현하기 어렵다는 단점이 있다. [6]에서는 비디오 데이터의 구조를 저장하기 위하여 기존의 데이터베이스 관리 시스템을 사용하고 비디오의 내용 정보는 키워드를 사용하여 주석 처리한다. 미리 정의된 함수를 사용하여 비디오 데이터들의 시간적 관계를 표현하고 비디오 데이터들을 공유한다.

[8]에서는 칼라 히스토그램, 색상 및 평균 밝기를 사용해서 비디오 스트림의 특성 정보를 자동적으로 추출하고 이 정보는 비디오 데이터를 검색하는데 필요한 키 프레임을 찾는데 사용한다. 한편, [9]에서는 비디오 데이터를 구조화하기 위해서 특정 신에 대한 비디오 정보를 정해진 모델링 요소 집합으로 구성된 벡터 표현식(vector expression)으로 변환된다. 이 벡터 표현식은 신의 경계를 감지하여 추출하기 위해서 이웃한 프레임과 세그먼트간의 유사성을 찾는데 사용한다.

OVID[11]와 Algebraic Video Model[12]에서는 비디오 데이터의 동적 정보를 유동적으로 정의된 속성과 속성값 쌍으로 기술한다. OVID에서의 기술 단위는 비디오 객체로서 내용을 기술하는데 사용된 모든 속성값들은 일반화 계층(generalized hierarchy) 구조에서 유지한다. 따라서 속성값을 많이 정의할 경우 트리의 크기가 커지는 단점이 있다. OVID에서는 고유의 질의 언어로 *VideoSQL*를 사용하여 비디오 객체를 검색한다. Algebraic Video Model에서의 기술 단위는 비디오 대수 연산자로 표현되는 비디오 표현식이며 이러한 구조는 다른 비디오 표현식에서 용이하게 공유할 수 있는 장점이 있지만 비디오 표현식이 대수 연산을 다중으로 포함할 경우 비디오 표현식에 해당하는 실제 비디오 데이터를 파악하는 관계가 복잡해진다. 또한 속성과 속성값 쌍의 논리 연산 조합으로 질의를 표현하지만 브라우징하기 위한 의미 사전 구조가 없기 때문에 사용자는 질의를 표현하기 위해 속성과 속성값의 리스트를 각각 순차적으로 검색해야한다.

기존의 DBMS에서 비디오 데이터의 정적 정보는 가장 효과적으로 표현할 수 있지만 대부분의 연구에서 내용 기반 질의를 표현하기 위해 고유의 질의 언어를 제공한다. 이들 질의 언어들은 SQL과 호환되지 않으므로 비디오 데이터의 질의를 동적 정보와 정적 정보의 조합으로 표현하기가 어렵다. 따라서, 이러한 방법은 새로운 질의 언어를 습득해야 하는 사용자의 부담을 초래할 뿐만 아니라 비디오 데이터베이스가 기존 데이터베이스와 통합되어야 할 경우 호환성의 문제가 발생할 수 있다.

두 개의 기술 단위 사이에는 다양한 시간적인 관계[13]가 존재할 수 있는데 이들 중에서 하나의 구간이

다른 구간에 포함될 때 구간 포함(interval inclusion) 관계가 발생하며 이 관계는 OVID 시스템에서 동적 기술을 상속하는데 사용한다. 속성은 사용자에 의해 상속 가능한 속성과 상속 불가능한 속성으로 분류되는데 상속 가능한 속성과 속성값 쌍은 병합(*merge*) 혹은 중첩(*overlap*)연산이 수행되었을 경우에만 상속된다. 반면에 Algebraic Video Model은 포함되는 비디오 표현식 간에 기술된 의미를 동적으로 상속할 수 있지만 OVID에서와 같이 선택적인 상속을 지원하지 않는다.

[14]에서는 비디오 데이터의 내용 기반 검색을 위해서 계층적 시간 언어(hierarchical temporal language)를 사용하고 비디오 데이터의 시공간적 기술은 논리식으로 표현한다. [15]에서는 멀티미디어 객체를 검색하기 위해 확률적 검색 모델(Probabilistic retrieval model)을 사용한다. 질의와 비디오 클립 내용간의 정확한 매칭에 따라 검색 결과를 제공하는 다른 방법과는 달리 이 방법은 질의와의 관련성을 확률로 표현한다. 비디오 데이터에 대한 모델링 요소는 비디오 데이터베이스의 응용 분야에 따라 다르게 정의될 수 있기 때문에 질의를 표현할 수 있는 범위는 모델링된 요소에 제한된다.

3. 유동적인 비디오 데이터 모델

3.1 속성의 유형

기존의 관계형 및 객체 지향 데이터베이스 모델에서는 비디오 데이터의 정적 정보는 손쉽게 표현할 수 있다[16,17]. 개체의 일반적인 성질에 연관된 속성들은 데이터베이스 설계 시에 고정 스키마로 정의한다. 예를 들어, 이름, 나이, 성별과 같은 개체의 일반적인 속성이나 제목, 파일 포맷, 크기, 해상도와 같은 비디오 파일 자체에 대한 정보는 고정 스키마로 손쉽게 모델링될 수 있다. 반면에 다양한 사용자가 서로 다른 관점으로 속성에 대한 정보를 여러 단계로 추상화시킬 경우 속성의 공통적인 집합을 정의하기가 불가능하기 때문에 동적 정보는 고정 스키마로 모델링하기가 어렵다. 따라서 OVID 시스템에서처럼 동적 정보는 속성과 속성값 쌍으로 표현한다. 속성과 속성값 쌍의 집합을 *동적 기술(dynamic description)*이라고 정의하고 그 요소는 필요에 따라 동적으로 추가 또는 삭제될 수 있다. 이런 유동적인 특성때문에 스키마를 정의하는 기존의 방법은 사용될 수 없으며 속성과 속성의 도메인을 명확히 정의하는 것이 용이하지 않기 때문에 유동적으로 정의된 속성과 속성값은 특별한 관리가 필요하다[21].

동적 정보는 형태 단계(*syntactic-level*)와 의미단계(*semantic-level*)의 두 가지 단계로 분류할 수 있다. 형태 단계 기술은 기술 단위에 나타나는 정보의 가장 하위 단계의 추상화이다. 이는 비디오데이터에 나타나는 개체의 크기, 위치 혹은 개체의 연속된 움직임의 방향으로서 개체의 정확한 모양, 색깔 또는 움직임을 표현하는데 사용한다. 따라서 형태 단계 기술은 대개 물리적으로 연속된 기술 단위에 사용된다. 기존의 많은 연구들은 형태 단계 정보만을 사용하여 비디오 데이터의 주석을 자동적으로 부여하는 기법에 대해 연구되었다[8,18-20]. 의미 단계 기술은 보다 상위 단계의 추상화로서 기술 단위에 나타나는 형태 단계 정보의 개념적인 의미를 모델링하는데 사용한다. 예를 들어 사람이 독서하고 있는 신이라면, 형태 단계 기술은 신에 나타나는 각 개체의 위치와 크기를 모델링할 수 있고, 의미 단계 기술은 신을 *독서*로 모델링할 수 있다. 형태 단계 기술과 달리, 의미 단계 기술은 여러 단계의 추상화가 가능하고 연속된 비트스트림 뿐만 아니라 하나 이상의 작은 기술 단위들이 하나의 큰 기술 단위로 정의될 수 있다. 이 경우에 하나로 구성되는 의미 단계 기술은 작은 기술 단위보다 높은 단계로 추상화 된 의미로 기술된다. 표 1은 비디오 데이터의 내용을 기술할 수 있는 속성형을 선행연구를 바탕으로 재정의하였다[21]. 동적 정보

표 1. 속성의 분류

고정 속성	물리 속성	비디오 데이터의 물리적 성질 (파일 이름, 위치, 크기, 해상도, 파일 포맷, 등등.)
	정적 속성	비디오 파일 혹은 뷰의 정적 정보 (제목, 제작자, 배우, 제작일, 요약, 등등.)
	개체 속성	응용 분야의 개체의 일반적인 성질 (이름, 나이, 가격, 크기, 등등.)
유동 속성	형태 속성	기술 단위의 개체 형태 (위치, 크기, 색깔, 방향, 등등.)
	의미 속성	기술 단위의 의미 (환경, 사건, 행동, 공간적 배경, 등등.)

를 표현하는 속성은 유동 속성이라고 정의하고, 그렇지 않은 경우를 고정 속성이라고 정의한다.

3.2 유동 속성 트리(Flexible Attribute Tree : FAT)

각 사용자의 관점에 따라 유동 속성과 속성값의 이름이 각각 다르게 표현할 수 있기 때문에 유동 속성과 속성값의 집합을 관리하는 것은 용이하지 않다. 예를 들어, 개체의 위치는 "위치" 혹은 "중심"으로 표현될 수 있고 그 속성값은 x-y 좌표 혹은 r-θ 좌표로 표현될 수 있다. 다양한 추상화 단계로 해석될 수 있는 유동 속성을 관리하기 위해 유동 속성 A에 대해서 *유동 속성 트리(flexible attribute tree : FAT)*를 정의한다. FAT T_A는 노드 N={A, n_1, n_2, …, n_p}의 집합으로 구성되고, 각 원소는 유일한 값 $\upsilon_i(1 \le i \le p)$를 갖는다. 또한 FAT T_A는 다음과 같은 성질을 갖는다.

(i) 노드 A는 트리의 루트이다.
(ii) 노드 n_i는 자신의 부모보다 상세화된 값 υ_i를 갖는다.
(iii) 노드 n_i가 내부 노드이면 n_i의 도메인은 $D(n_i)=\upsilon_i$이다.
(iv) 노드 n_i가 리프 노드이면,
- n_i는 사용자에 의해 정의되는 상세화된 값 W={w_1, w_2, …, w_i}의 집합을 가질 수 있다
- n_i의 도메인 $D(n_i)=\{\upsilon_i\} \cup W$이다.
(v) T_A의 도메인 $D(T_A)=\cup_{i=1}^{p} D(n_i)$이다.
(vi) 새로운 노드가 트리에 삽입될 수 있고 트리의 노드는 삭제될 수 있다. 노드 n_i가 삭제되었을 때는 노드 n_i의 자식은 노드 n_i 부모의 자식이 된다.

FAT에 있는 노드의 값으로 고유 명사와 같은 특정값은 사용할 수 없는 대신에 리프 노드에서 상세화된 값으로 관리한다. 예를 들어 FAT *빌딩*과 그 리프 노드의 하나가 *정부 기관*이라면, *청와대*는 정부 기관의 상세화된 값이 된다. 그림 1에서는 의미 속성 "행위"의 FAT와 x-y 좌표상에 나타나는 개체의 위치를 표현하는 형태 속성 "위치"의 FAT를 보여준다.

동일한 개념의 트리를 두개의 다른 FAT 또는 속성값으로 모델링할 수 있기 때문에 FAT와 노드 값을 일관되게 사용하기 위해서 관리자만이 새로운 FAT를 정의할 수 있고 FAT의 구조를 변경할 수 있다. 주어진 FAT에 기초하여 사용자는 신에 대한 주석을 부여할 수 있고 내용 기반 질의를 표현할 수 있다.

3.3 비디오 클립

앞서 기술하였듯이 비디오 데이터베이스에서는 모델링되는 다양한 유형의 정보가 존재한다. 기존 데이터베이스 모델에서 이러한 정보들을 나타내기 위해서 다음과 같은 스키마 구조를 제안한다.

- 물리세그먼트(<u>P_{id}</u>, 파일이름, 파일위치, 파일크기, 파일포맷,… *물리속성 or 정적속성*)
- 논리세그먼트(<u>L_{id}</u>, P_{id}, 시작옵셋, 끝옵셋)
- 논리세그먼트리스트(<u>V_{id}, L_{id}, Sequence#</u>)
- 비디오클립(<u>V_{id}</u>, 유동속성이름, 속성값, 상속정보?, [*개체속성이름*])
- 뷰(<u>V_{id}</u>, *고정속성*)
- 유동속성(<u>유동속성이름</u>, 유형, 상속여부?, FAT pointer)

비디오 데이터베이스에 삽입되는 각 비디오 파일은 유일한 물리세그먼트 P_{id}를 가지며 논리세그먼트

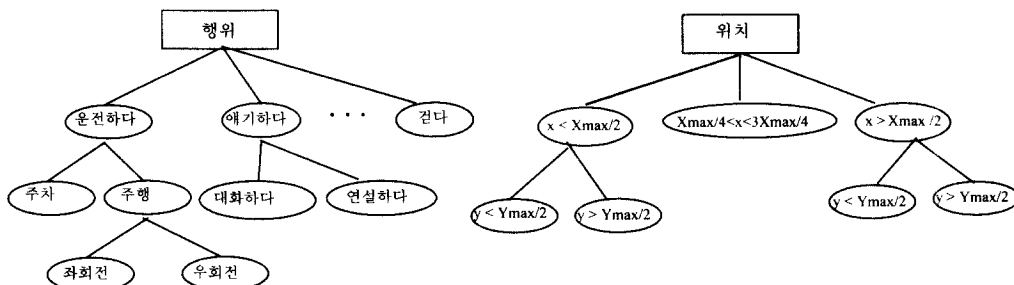



그림 1. FAT의 예

$L_{id}=(P_{id},[s,e])$는 물리세그먼트 P_{id}에 있는 옵셋 s에서 옵셋 e까지의 구간을 나타내는 것으로 정의한다. 다수의 논리세그먼트가 하나의 물리세그먼트에서 정의될 수 있는 반면에, 그 반대의 경우는 정의할 수 없기 때문에 물리세그먼트와 논리세그먼트의 집합 간의 관계는 1:N이다. 또한 하나 이상의 논리세그먼트들을 *논리세그먼트리스트* $[(P_1,[s_1,e_1]),\ (P_2,[s_2,e_2]),\ ...,\ (P_n,[s_n,e_n])]$로 정의하여 *비디오 클립* 혹은 *뷰*를 정의하는데 사용한다. 비디오 클립은 내용 기반 의미를 기술하는 단위인 반면에, 뷰는 논리적인 비디오 파일을 구성하기 위해 사용한다. 즉 뷰는 여러 물리세그먼트들의 부분을 논리적으로 연결하여 하나의 논리적인 비디오 파일을 구성하는데 사용된다. 비디오 클립 혹은 뷰는 하나 이상의 논리세그먼트로 정의될 수 있으며 동일한 논리세그먼트는 여러 뷰 혹은 비디오 클립에서 공유할 수 있으므로 비디오 클립 집합 또는 뷰 집합은 논리 세그먼트 집합과 M:N관계를 형성한다.

비디오 클립의 동적 의미는 FAT에 있는 노드값을 선택하여 표현하며 실제의 비디오 클립에 연결된다. 사용자는 비디오 클립을 생성하고 제거할 수 있지만 논리세그먼트와 논리세그먼트리스트는 시스템에서 유지된다. *물리세그먼트*와 *뷰* 스키마의 고정 속성과 *비디오 클립* 스키마의 개체 속성은 비디오 데이터베이스의 응용 분야에 따라 정의된다. 대개 개체 속성은 속성과 속성값 쌍의 주체로서 개체의 이름을 표현한다. 이는 비디오 클립에서 하나 이상의 개체들이 동시에 나타날 수 있기 때문에 각 개체들을 독립적으로 기술할 수 있도록 지원한다.

비디오 클립 υ_i와 υ_j가 각각 $\upsilon_i = [l_1, l_2\cdots, l_x]$와 $\upsilon_j = [l'_1, l'_2\cdots, l'_y]$으로 정의되는 논리세그먼트리스트라고 하고, υ_j의 비디오 스트림이 υ_i의 모든 스트림을 포함하는 것을 $\upsilon_i \angle \upsilon_j$으로 나타낸다면 υ_i에 있는 각 논리세그먼트 $l_i = [p_i, (s_i, e_i)]$에 대해, $\upsilon_i \angle \upsilon_j$은 다음의 (i)과 (ii)를 만족하는 논리 세그먼트 $l'_j = [p'_j, (s'_j, e'_j)]$가 υ_j 에 존재해야 한다.

(i) $p_i = p'_j$ 그리고
(ii) $s'_j < s_i,\ e_i < e'_j$

주어진 형태 속성 집합 A_{syn}과 의미 속성 집합 A_{sem}에 대하여, 비디오 클립 υ_i에 대해 기술된 의미는 다음과 같이 각각 υ_i^{syn} 과 υ_i^{sem} 라고 정의한다.

$$\upsilon_i^{syn} = \{\ (\text{속성, 속성값}) \mid \text{속성} \in A_{syn},\ \text{속성값} \in D(T_{attr})\ \}$$
$$\upsilon_i^{sem} = \{\ (\text{속성, 속성값}) \mid \text{속성} \in A_{sem},\ \text{속성값} \in D(T_{attr})\ \}$$

따라서 비디오 클립 υ_i 에 대해서 사용자가 기술한 의미는 다음과 같다.

$$\upsilon_i^{self} = \upsilon_i^{syn} \cup \upsilon_i^{sem}$$

구간 포함 관계 연산에 의한 기술을 고려하기 위해서 의미 속성 A_{sem}의 집합을 다음과 같이 상속 가능 속성(A_{I_sem})과 상속 불가능 속성(A_{NI_sem})으로 분류할 수 있다.

$$A_{sem} = A_{I_sem} \cup A_{NI_sem},\ A_{I_sem} \cap A_{NI_sem} = \varnothing$$

스키마 관리자는 임의의 의미 속성이 상속 가능한지, 상속 불가능한 지를 결정한다. 따라서 비디오 클립 υ_i 의 의미 기술은 다음과 같이 상속 가능 기술($\upsilon_i^{I_sem}$)과 상속 불가능 기술($\upsilon_i^{NI_sem}$)의 두 가지로 분류된다.

$$\upsilon_i^{I_sem} = \{\ (\text{속성, 속성값}) \mid \text{속성} \in A_{I_sem},\ \text{속성값} \in D(T_{attr})\ \}$$
$$\upsilon_i^{NI_sem} = \{\ (\text{속성, 속성값}) \mid \text{속성} \in A_{NI_sem},\ \text{속성값} \in D(T_{attr})\ \}$$

비디오 클립에 대한 모든 형태 기술은 상속될 수 있으므로 비디오 클립 υ_i 에 상속된 기술은 다음과 같이 $\upsilon_i \angle \upsilon_k$ 을 만족하는 모든 υ_k의 상속가능 기술을 포함한다.

$$\upsilon_i^{inh} = \cup_k\ (\upsilon_k^{syn} \cup \upsilon_k^{I_sem})$$

따라서 상속된 기술을 포함하는 비디오 클립 υ_i 에 기술된 의미는 다음과 같다.

$$\upsilon_i^{description} = \upsilon_i^{self} \cup \upsilon_i^{inh}$$

모든 유동 속성의 이름, FAT, 그리고 동적 정보의 유형(형태 혹은 의미)은 *유동 속성* 스키마에 저장되고 속성의 상속 여부를 유지한다. 비디오 클립간의 의미 상속은 *비디오 클립* 스키마에서 표현된다. 사용자가 비디오 클립 υ_s에 표현한 새로운 형태 기술이나 상속 가능한 의미 기술은 *비디오 클립* 스키마에 다음과 같이 표현된다.

(v_s, 속성′, 속성값′, 상속정보=no,[la])
where *(속성′, 속성값′)* ∈ {(속성,속성값) | 속성 ∈ A_{syn} or 속성 ∈ A_I, 속성값 ∈ $D(T_{attr})$} 그리고
[la] : 개체 속성값

이때 $v_t \angle v_s$를 만족하는 모든 비디오 클립 v_t에 대하여 새로운 튜플 *(v_t, 속성′, 속성값′, 상속정보=yes,[la])*을 *비디오 클립* 스키마에 부가적으로 첨가하게 된다. 포함(∠) 관계를 만족하는 비디오 클립이 많다면 의미 상속 처리 시간은 많이 소요되지만 상속된 기술은 비디오 클립의 보충적인 의미로 고려할 수 있기 때문에 의미 상속 처리는 새로운 기술이 삽입될 때마다 수행하기 보다는 주기적으로 수행하는 것이 더욱 효율적이다.

의미 기술을 상속할 때 부분 중첩 관계(*partially overlapped relationship*)는 구간을 부분적으로 공유하고 있는 두개의 클립간에 발생하며 이 관계는 중첩된 부분 구간이 각각의 비디오 클립과 포함 관계를 가지므로 특별한 구간 포함 관계로 간주할 수 있다. 따라서 중첩된 부분을 새로운 비디오 클립으로 시스템이 정의함으로써 어느 사용자에 의해서도 비디오 클립으로 정의되지 않은 부분에 대해 내용 기반 검색을 지원할 수 있다.

4. 비디오 클립에 대한 질의

단순한 내용 기반 질의는 그래픽 사용자 인터페이스(GUI)로 표현할 수 있지만 복잡한 질의는 비디오 데이터 모델에 기반한 질의 언어로 표현해야 한다. 기존 연구에서 사용된 대부분의 질의 언어는 세가지 방식으로 분류된다. 첫번째는 질의를 표현하는데 형식화된 방법을 제공하지 못했던 임기 응변적인 방법(ad-hoc solution)을 사용한 것으로 키워드 형태로 주석을 부여하는 시스템에서 사용한다. 두번째는 SQL과 같은 표준 질의 언어를 사용한 것이다. 그러나 이 방법은 스키마가 유동적으로 바뀔 수 없기 때문에 모델링 요소가 고정되어 있는 분야에만 한정되었다. 세번째는 제안하는 시스템에 적합한 형태를 갖는 고유의 질의 언어를 제공한 것이다. 본 장에서는 3장에서 제안한 스키마 구조에 대해 내용기반 검색에 필요한 질의 형태를 기존의 데이터베이스 관리 시스템에서 사용하는 SQL언어로 표현하는 방법을 제시한다.

비디오 데이터베이스에 정치인 비디오 파일이 저장되어 있다고 가정하자. FAT가 그림 1과 같다고 가정할 때 김영삼 대통령이 연설하고 있는 비디오 클립을 찾고자 한다면 질의는 다음과 같은 SQL로 표현된다.

Q1: select V_{id}
from *비디오 클립*
where 유동속성=행위 and 값=연설and 개체이름=김영삼

사용자가 화면의 $x<X_{max}/2$와 $y<Y_{max}/2$의 범위에서 김영삼이 나타나는 비디오 클립을 모두 찾고자 한다면 질의는 다음과 같다.

Q2: (select V_{id}
from *비디오 클립*
where 유동속성=위치 and 값=$x< X_{max}/2$ and 개체속성=김영삼)
intersect
(select V_{id}
from *비디오 클립*
where 유동속성=위치 and 값= $y<Y_{max}/2$ and 개체이름=김영삼)

관계형 데이터베이스에서 스키마 *정치인(이름, 나이, 성별, 국적, 활동경력)*이 존재한다고 할 때 한국 남자 정치가가 연설하는 클립을 검색하는 질의는 다음과 같이 표현할 수 있다.

Q3: select V_{id}
from *비디오 클립*
where 유동속성=행위 and 값=연설 and
개체이름 in (select 이름
from 정치인
where 국적=한국 and 성별=남자)

비디오 클립간의 포함*(∠)* 관계를 파악하기 위해 한 구간이 다른 구간에 포함되는지를 확인할 필요가 있다. 예를 들어, 질의 Q4가 물리 세그먼트 p의 t_1에서 t_2까지의 구간에서 정의된 모든 논리세그먼트를 검색하고, Q5가 비디오 클립 v의 모든 논리세그먼트를 검색한다면 질의는 다음과 같다.

```
Q4: select L_id
    from 논리세그먼트
    where P_id = p and 시작옵셋 > t1 and 끝옵셋
    < t2

Q5: select L_id
    from 논리세그먼트리스트
    where V_id = v
    order by Sequence#
```

상속된 기술을 제외하고 비디오 클립을 검색하고자 한다면 *비디오 클립* 스키마에 있는 *상속정보?* 속성을 이용하여 표현한다. 예를 들어, 질의 Q2를 상속된 기술이 없는 질의로 표현하면 다음과 같다.

```
Q6: (select V_id
    from 비디오 클립
    where 유동속성=위치 and 값=x<X_max /2 and
    개체속성=케네디 and 상속정보?=no)
    intersect
    (select V_id
    from 비디오 클립
    where 유동속성=위치 and 값=y<Y_max /2 and
    개체이름=케네디 and 상속정보?=no)
```

비디오 클립의 수와 클립에 대한 기술이 방대해진다면 클립 검색 작업은 많은 탐색 시간이 필요하지만 OVID나 Algebraic Model에서는 이를 위한 접근 구조에 대한 해결책이 제시되지 않았다. 이와 같이 기존의 데이터베이스 관리 시스템을 이용할 때 얻을 수 있는 장점은 해쉬(hash)나 B-tree와 같은 접근 구조가 데이터베이스 검색 효율을 높일 수 있다는 것이다.

5. 비디오 브라우징

내용 기반 질의는 속성에 대한 다양한 조건을 포함할 수 있기 때문에 비디오 데이터베이스에서 사용자는 복잡한 질의를 표현하기가 힘들며 속성과 속성값을 유동적으로 관리하게 되면 이 문제는 더 심각해질 수 있다. 또한 사용자가 필요한 모든 질의 조건을 구성했다고 해도 사용자는 질의의 결과 비디오를 사용하기 전에 반드시 재생해야 한다. 따라서 이 장은 사용자가 질의를 표현하고 결과를 분석하는데 도움을 줄 수 있는 브라우징 방법에 대해 기술한다.

5.1 사전 브라우징

FAT의 구조는 직접적으로 사용자가 사용하기에 적합하지 않다. 노드가 추가되고 삭제되기 때문에 FAT는 균형이 맞지 않는 트리가 되기 쉬우므로 따라서 전체적인 트리 구조를 보여주기 어렵다. 또한, 하나의 FAT가 속성과 속성값 쌍을 표현하기 때문에 복잡한 질의 조건을 구성하기 위해서 대부분의 질의는 하나 이상의 FAT가 필요하다.

그림 2는 FAT에 있는 속성값의 브라우징 방법을 보여준다. 하나의 FAT에서 관련된 속성값을 보여주기 위해서 FAT의 루트에서부터 현재 선택된 노드의 경로에 이웃한 노드가 사용자의 작업 대상이 된다. 사용자가 트리의 탐색 경로를 선택할 때 브라우저는 경로에 있는 동일한 레벨의 속성값만을 표시한다. 브라우저에서 속성값 상자는 FAT에 있는 노드의 속성값을 표현하고 FAT의 레벨 차이는 각 레벨마다 속성값 상자의 색상을 다르게 함으로써 표현한다. 속성값 상자의 두 가지 화살표는 부모 혹은 자식으로 이동할 수 있는 콘트롤로 그림 2의 속성값 상자의 자식 콘트롤을 선택하면 속성값 상자는 자신의 자식 노드에 해당하는 속성값 상자로 대치되고 속성값 상자의 부모 콘트롤을 선택하면 자식 콘트롤과 반대의 기능을 한다.

본 논문에서 제안하는 브라우저에서는 하나 이상의 FAT를 그림 3과 같이 동시에 브라우징할 수 있다. 그림 2는 세 개의 FAT A^i, A^j, A^k 를 하나의 작업 공간에서 검색하는 모습을 나타낸 것으로 브라우저에서 유동 속성을 나타내는 각 열을 *속성 팔레트 (attribute pallete)*라고 정의한다. 초기 화면에서 아이콘 형식으로 제공된 모든 팔레트의 이름중 사용자가 검색하고자 하는 아이콘을 선택하면 해당 팔레트가 브라우저에 나타난다.

관리자는 FAT의 구조를 수정할 수 있는데 FAT의 부분 트리(sub-tree)의 크기가 커지면 *비디오 클립* 스키마의 내용을 수정하여 새로운 FAT로 정의하여 관리한다. 즉 *비디오 클립* 스키마에서 새로운 FAT로 정의하고자 하는 서브 트리의 속성값을 갖는 모든 튜플들의 *유동 속성* 필드를 새로운 FAT의 이름으로 변환시킨다. 또한 FAT의 수가 관리하기 힘들 정도로 커지면 의미의 유사성에 따라 클립맵 [7]

(a) FAT

(b) A^i를 선택한 후

(c) v_7 의 자식 콘트롤을 선택한 후

그림 2. FAT의 브라우징

그림 3. 사전 브라우저

과 같이 부 그룹화(sub-grouping)할 수 있다.

5.2 비디오 클립 브라우징

FAT를 브라우징하면서 사용자는 비디오 클립의 의미를 기술할 수 있다. FAT에서 원하는 유동 속성값을 선택한 후에 *속성=속성값*의 기본적인 질의 조건을 입력할 수 있는 속성값 상자를 클릭하여 의미를 기술할 수 있는데 사용자는 그림 2-(d)의 질의 조건을 나타내는 텍스트 상자에서 리프 노드의 상세 속성값을 입력하고 유사한 방법으로 개체 속성값을 입력함으로써 비디오 클립 정보를 *비디오 클립* 스키마에 저장한다.

비디오데이타에 대한 질의를 표현하는 기존의 방법에서 상호 대화식 질의는 키워드의 리스트[5,6] 혹은 논리 연산자[4,7,8,12]의 조합으로 구성하였다. 이들은 질의 조건을 제한하거나[7] 색깔, 질감(texture), 공간/시간적 조건[4,7]을 지원하기 위한 고유의 인터페이스를 제공하였다. 본 논문에서 질의를 할 때에는 5장 1절에서 기술된 사전 브라우저를 사용하여 표현할 수 있으며 상세 속성값에 대한 질의 조건은 앞서 기술한 의미 기술 방법과 동일한 방법으로 표현할 수 있다. 이렇게 구성된 질의 조건은 논리 연산으로 해석되어 내부적으로 SQL문으로 변환되며 동일한 팔레트에서 선택한 질의 조건은 속성값간의 OR로 다른 팔레트에서 선택한 질의 조건들은 AND로 나타내어진다. 그러므로 동일한 FAT의 값들간에 AND 조건을 표현하기 위해서는 동일한 속성 팔레트를 한 번 이상 선택해야 된다.

사전 브라우저의 전반적인 구조는 그림 3과 같다. 그림 3은 4개의 고정 속성과 5개의 유동 속성을 선택하여 질의를 구성한 후의 브라우저를 보여준다. 두개의 유동 속성 및 속성값 쌍, *ACTION/handshake*와 *POSITION/>Xmax/2*의 주체는 *김영삼 대통령*으로 표현되었고 *PLACE* 속성에 대하여 리프 노드 *government organization*의 상세 속성값은 *Blue House*이다. 여기에 각 질의 조건마다 제공되는 *Inh* 버튼을 선택하여 사용자는 상속된 의미를 포함하는 질의 결과를 얻을 수 있다. 질의 결과는 그림 4와 같이, *결과 뷰어*에 나타나며 주어진 질의를 만족하는 각 비디오 클립의 대표 프레임이 아이콘으로 표현된다. 이 아이콘은 [7]에 있는 마이콘(micon)과 동일하며 사용자는 대표 프레임에 해당되는 비디오 데이터를 재생하기 위해서 아이콘을 클릭할 수 있고 슬라이드 쇼 기능을 사용하여 모든 대표 프레임을 순차적으로 짧은 시간에 볼 수 있다.

[5]에서의 질의 결과는 매칭 스코어를 순차적으로 정렬한 리스트로 표시하고 뉴스 방송에서 시간바(timebar) 구조를 제공하여 질의 결과에 대한 매칭 정도를 표현한다. 하지만 비디오 데이터베이스에서는 사용자가 질의를 표현하는데 도움을 줄 수 있도록 좀더 분석적인 기능을 제공할 필요가 있기 때문에 질의 결과가 너무 많거나 적으면 사용자는 적합한 검색 결과를 얻기 위하여 질의 조건을 재구성할 필요가 있다. 결과의 수는 질의의 구성뿐만 아니라 데이터베이스의 내용에 의존되므로 사용자가 구성한 질의 조건의 부분적인 예상 결과 수를 보여줄 수 있는 특별한 브라우저가 필요하다. 그림 5는 본 논문에서 제안한 *결과 브라우저*의 모습으로 s_i, $1 \le i \le c$를 사용자가 표현한 "속성=속성값"으로 표현되는 질의 조건이라 할 때 질의의 부분적인 결과 분석을 위해서 *부*

그림 4. 결과 뷰어

그림 5. 결과 브라우저

*분 결과 행렬(partial result matrix)*이라고 불리는 CxC 행렬 R을 사용한다. 이 행렬의 각 원소는 질의 조건의 조합을 만족하는 비디오 클립의 수를 다음과 같이 나타낸다.

(i) 대각 원소(Diagonal elements) : R(i, i), $1 \le i \le c$: s_i 를 만족하는 비디오 클립의 수

(ii) 상위 대각 원소(Upper-diagonal elements) : R(i,j), $1 \le i < c$, $i < j \le c$: $s_i \wedge s_{i+1} \wedge \cdots \wedge s_j$를 만족하는 비디오 클립의 수

(iii) 하위 대각 원소(Lower-diagonal elements) : R(i,j), $1 < i \le c$, $1 \le j < i$: $s_i \wedge s_j$를 만족하는 비디오 클립의 수

직관적으로 해석할 수 있도록 부분 결과 행렬의 각 원소의 색상 명도로 결과 비디오 클립수를 나타낸다.

위의 질의 조건들의 조합은 사용자가 질의의 매칭 정도를 이해할 수 있도록 충분한 정보를 제공하는데 행렬에서 원소의 순서가 질의의 결과를 분석하는데 중요한 요소이므로 사용자가 드래그 앤 드롭하여 재정렬할 수 있도록 지원한다. 결과 브라우저의 비디오 클립수에 기초하여 사용자는 질의를 재구성할 수 있다.

하나의 비디오 파일에 대한 물리적인 구조의 브라우징을 지원하기 위하여 *구조 브라우저(structure browser)*를 제공하며 브라우저의 구성은 그림 4의 결과 뷰어와 유사하다. 구조 브라우저에서 2차원 구조는 샷(shot), 신(scene), 시퀀스(sequence)의 대표 프레임을 나타내는 3-단계의 트리 구조로 표현하고 이 기능은 [7]에서의 계층적인 브라우저와 유사하다고 보면 된다. 즉 트리는 비디오 파일의 물리적인 구조를 표현하고 있으며 구조 브라우저에서는 비디오 파일에 정의되어 있는 비디오 클립을 검색할 수 있다.

6. 결 론

비디오 데이터의 특성상 비디오 데이터베이스에서 내용 기반 검색을 지원하는 방법은 간단하지 않다. 본 논문은 비디오 데이터의 복합적인 의미를 표현하는 데 필요한 속성의 유형을 선행연구[21]을 기반으로 재정의하였으며 이런 속성을 기존 데이터베이스 모델에서 표현하기 위해 두 가지 조합된 방법을 제안하였다. 하나는 서로 관련된 속성들을 그룹화하여 고정 스키마를 정의하는 것이고 다른 하나는 고정 스키마없이 속성을 정의하기 위해 유동적인 방법을 지원하는 것이다. 비디오 신이 여러 단계의 추상화된 의미로 해석될 수 있기 때문에 사용자가 신을 유동적으로 기술할 수 있도록 지원할 수 있어야 하며 다른 사용자가 기술한 내용도 일관되게 유지될 수 있어야 한다. 본 논문에서 제안하는 FAT는 이런 요구들을

만족하고 유동적으로 정의된 속성은 고정 스키마로 유지되기 때문에 표준 SQL로도 표현할 수 있다. 따라서 사용자가 내용 기반 질의를 고정 속성 및 유동 속성 등의 다양한 속성 조합으로 질의를 구성할 수 있다.

비디오 데이터와 비디오 클립간의 매핑 관계는 비디오 파일의 공통적인 부분을 재사용할 수 있도록 하며 이는 방대한 양의 비디오 파일이 저장되는 비디오 데이터베이스에서 중요한 의미를 차지한다. 따라서 비디오 클립에 대한 의미 기술은 구간 포함 관계와 부분 중첩 관계에 의해 공유되어 의미 기술을 상속하고 새로운 비디오 클립으로 정의할 수 있다. 또한 본 논문에서는 내용기반 검색을 효과적으로 지원할 수 있는 몇 가지 브라우징 방법을 제안하였다. 사전 브라우저는 내부적으로 트리 형태로 구조화된 FAT의 브라우징을 단순화시키고 구조 브라우저는 비디오 파일의 내용을 이해할 수 있도록 체계적인 방법을 제공한다. 부분 결과 행렬을 이용하여 질의를 재구성할 수 있도록 지원하는 결과 브라우저는 사용자가 질의의 조건들을 다양하게 조합하여 예상되는 질의 결과를 분석할 수 있도록 함으로써 비디오 데이터의 내용기반 검색의 효율을 높인다.

참 고 문 헌

[1] R. Price, "MHEG:An Introduction to the future International Standard for Hyermedia Object Interchange," *ACM proceedings of Multimedia*, CA, USA. 1993.

[2] D. Delodderе, W. Verbiest, and H. Verhille, "Interactive Video On Demand," *IEEE Communications Magazine*, May 1994.

[3] D.L. Gall. "MPEG: A Video Compression Standard for Multimedia Applications," *Communication of ACM*, April 1991.

[4] T. Kuo and A. Chen, "A Content-Based Query Language for Video Databases," *IEEE Proceedings of Multimedia*, 1996.

[5] M.G. Brown et al., "Automatic Content-based Retrieval of Broadcast News", *ACM Proceedings of Multimedia '95*, San Francisco, CA. USA, 1995.

[6] R. Hjelsvold and R. Midtstraum, "Modeling and Querying Video Data," *Proceedings of 20th VLDB Conference*, Santiago, Chile, 1994.

[7] S. Smoliar and H. Zhang, "Content-based Video Indexing and Retrieval," *IEEE multimedia*, summer 1994.

[8] H.J. Zhang, C.Y. Low, S.W. Smoliar and J.H. Wu, "Video Parsing, Retrieval and Browsing: An Integrated and Content-based Solution," *ACM proceedings of Multimedia '95*, San Francisco, CA USA 1995.

[9] M. Shibata and Y. Kim, "Content-based Structuring of Video Information", *IEEE Proceedings of Multimedia*, 1996.

[10] A. Ono, M. Amano, and M. Hakaridani, "A Flexible Content-Based Image Retrieval System with Combined Scene Description Keyword," *IEEE proceedings of Multimedia*, 1996.

[11] E. Oomoto and K. Tanaka, "OVID: Design and Implementatio of a Video-Object Database System," *IEEE trans. On Knowledge and Data Engineering*, Vol.5, No.4, August, 1993.

[12] R. Weiss, A. Duda and D. Gifford, "Composition and Search with a Video Algebra," *IEEE Multimedia*, Spring 1995.

[13] T. D. Little and A. Ghafoor, "Interval-Based Conceptual Models for Time-Dependent Multimedia Data," *IEEE trans. On Knowledge and Data Engineering*, Vol.5, No.4, August 1993.

[14] A. P. Sistla, C. Yu and R. Venkatasubrahmanian, "Similarity Based Retrieval of Videos", *ACM International Conferenece on Data Engineering*, April 1997.

[15] M. Ortega et. al, "Supporting Similarity Queries in MARS", *International Multimedia Conference*, Nov. 1997.

[16] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, Benjamin/Cummings Publishing Company, 1994.

[17] R.G.G. Cattell, Object Database Standard: ODMG-93, Morgan Kaufmann Publishers,

Inc. 1994.

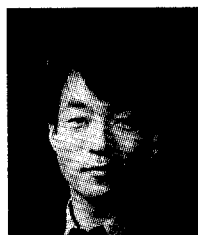
[18] S. Chang et al., "VideoQ:An Automated Content Based Video Search System using Visual Cues", *ACM International Multimedia Conference*, Nov. 1997.

[19] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki, "Structured Video Computing," *IEEE Multimedia*, fall 1994.

[20] Y. Chang et. al, "Integrated Image and Speech Analysis for Content-based Video Indexing", *IEEE Proceedings of Multimedia*, 1996.

[21] 정미영, 이원석 "효율적인 의미검색을 위한 동영상 데이터 모델링", *한국정보처리학회* 제 4권 제 4호, 1997년 4월

성 인 용

1997년 연세대학교 수학과 졸업
1999년 연세대학교 컴퓨터과학과 석사
1999년~현재 동양 시스템즈 KM 사업본부 근무
관심분야 : 동영상 브라우저, 멀티미디어, 검색 엔진 및 로봇임

이 원 석

1985년 Boston University 컴퓨터공학과 졸업
1987년 Purdue University, 컴퓨터공학 석사
1990년 Purdue University, 컴퓨터공학 박사
1990년 5월~1992년 12월 삼성전자 컴퓨터부문 선임연구원
1993년~현재 연세대학교 컴퓨터과학과 부교수
관심분야 : 비디오 데이터 모델링, 멀티미디어 데이터베이스, 데이터 마이닝임